
Diffusion-Driven Progressive Target Manipulation for Source-Free Domain Adaptation

Yuyang Huang^{1*}, Yabo Chen^{1*}, Junyu Zhou¹, Wenrui Dai^{1†}, Xiaopeng Zhang^{2†},
Junni Zou^{1†}, Hongkai Xiong¹, Qi Tian²

¹Shanghai Jiao Tong University, Shanghai, China ²Huawei Inc., Shenzhen, China
{huangyuyang, chenyaobo, blabla, daiwenrui, zoujunni, xionghongkai}@sjtu.edu.cn
zxphistory@gmail.com, tian.qi1@huawei.com

*These authors contributed equally to this work. Corresponding authors: Wenrui Dai; Xiaopeng Zhang; Junni Zou.

Appendix

A Evaluation Datasets

We evaluate our method on four Source-free Domain Adaptation (SFDA) datasets benchmarks, including the small-scale Office-31 dataset [25], the medium-scale Office-Home dataset [31], and two large-scale datasets (*i.e.*, VisDA [21] and DomainNet-126 [20]). The complete dataset statistics and domain configurations are elaborated on below.

Office-31 [25]: This small-scale dataset contains 31 object categories commonly found in office environments (e.g., keyboards, laptops, file cabinets). It comprises 4,652 images across 3 domains: *Amazon*, *Webcam*, and *DSLR*.

Office-Home [31]: Office-Home is a more challenging medium-scale domain adaptation dataset with 65 categories and 15,500 images distributed across 4 domains: *Art*, *Clipart*, *Product*, and *Real-World*.

VisDA [21]: VisDA is a large-scale dataset for domain adaptation, originally designed for the 2017 Visual Domain Adaptation Challenge. It focuses on synthetic-to-real transfer with 12 object categories. The source domain contains 152,397 synthetic images, while the target domain has 55,388 real-world images.

DomainNet-126 [20]: DomainNet-126 is a subset of the full DomainNet dataset [20], curated for domain adaptation research. It includes 126 object categories across 4 domains: *Clipart*, *Painting*, *Real*, and *Sketch*. The dataset contains 145k images, making it one of the largest SFDA benchmarks.

B Implementation Details

B.1 Source Model Pre-training

For the source model training, we employ only the cross-entropy loss as the objective function. Regarding the training hyperparameters, we set `weight_decay` (weight decay, L2 regularization coefficient) to $5e-4$, `lr_gamma` (learning rate gamma, learning rate scheduler parameter) to 0.0003, `lr_decay` (learning rate decay rate) to 0.75, and momentum (SGD momentum parameter) to 0.9. For the large-scale DomainNet-126 and VisDA, to facilitate the model convergence, we set batch size as 128, `n_iter_per_epoch` (Number of iterations per epoch) as 200, `n_epoch` (Total training epochs) as 100, `lr` (learning rate) as $3e-3$. For other datasets, we set batch size as 32, `n_iter_per_epoch` as 300, `n_epoch` as 50, and `lr` as $1e-3$. As for source model architecture, we employ ResNet-50 for Office-31 [25], Office-Home [31] and DomainNet-126 [20], and ResNet-101 for VisDA [21].

Table A.1: Full Results (%) on **VisDA** evaluated with ResNet-101. The top three performances in each column are highlighted in red, orange, and yellow, respectively.

Method	Venue	VisDA													
		plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Perclass	
Source	–	92.3	33.3	76.4	60.9	86.5	32.7	89.9	33.3	79.8	48.0	87.7	18.4	61.6	
CPGA [22]	IJCAI21	95.6	89.0	75.4	64.9	91.7	97.5	89.7	83.8	93.9	93.4	87.7	69.0	86.0	
ASOGE [6]	TCSVT23	94.9	84.3	76.8	54.3	94.9	93.4	86.0	85.0	87.2	90.0	86.7	62.7	83.2	
ISFDA [19]	CVPR24	97.5	91.4	87.9	79.4	97.2	97.2	92.2	83.0	96.4	94.2	91.1	53.0	88.4	
PS [10]	ML24	95.3	86.2	82.3	61.6	93.3	95.7	86.7	80.4	91.6	90.9	86.0	59.5	84.1	
DM-SFDA [5]	–	98.1	89.8	90.6	90.5	96.8	95.2	92.2	93.4	97.8	94.4	92.4	48.8	86.3	
SHOT [15]	ICML20	95.0	87.4	80.9	57.6	93.9	94.1	79.4	80.4	90.9	89.8	85.8	57.5	82.7	
NRC [33]	NIPS21	96.8	91.3	82.4	62.4	96.2	95.9	86.1	90.7	94.8	94.1	90.4	59.7	85.9	
GKD [27]	IROS21	95.3	87.6	81.7	58.1	93.9	94.0	80.0	80.0	91.2	91.0	86.9	56.1	83.0	
AaD [32]	NIPS22	97.4	90.5	80.8	76.2	97.3	96.1	89.8	82.9	95.5	93.0	92.0	64.7	88.0	
AdaCon [4]	CVPR22	97.0	84.7	84.0	77.3	96.7	93.8	91.9	84.8	94.3	93.1	94.1	49.7	86.8	
CoWA [12]	ICML22	96.2	89.7	83.9	73.8	96.4	97.4	89.3	86.8	94.6	92.1	88.7	53.8	86.9	
SCLM [30]	NN22	97.1	90.7	85.6	62.0	97.3	94.6	81.8	84.3	93.6	92.8	88.0	55.9	85.3	
ELR [35]	ICLR23	97.1	89.7	82.7	62.0	96.2	97.0	87.6	81.2	93.7	94.1	90.2	58.6	85.8	
PLUE [18]	CVPR23	94.4	91.7	89.0	70.5	96.6	94.9	92.2	88.8	92.9	95.3	91.4	61.6	88.3	
CPD [36]	PR24	96.7	88.5	79.6	69.0	95.9	96.3	87.3	83.3	94.4	92.9	87.0	58.7	85.5	
TPDS [26]	IJCV24	97.6	91.5	89.7	83.4	97.5	96.3	92.2	82.4	96.0	94.1	90.9	40.4	87.6	
DIFO [29]	CVPR24	97.6	88.7	83.7	80.8	95.9	95.3	91.9	85.0	89.4	93.2	93.2	69.0	88.6	
ProDe [28]	ICLR25	96.6	90.3	83.9	80.2	96.1	96.9	90.3	86.4	90.8	94.0	91.3	67.0	88.7	
DPTM (ours)	–	99.5	97.1	96.2	93.0	99.2	99.2	98.8	97.7	99.3	99.7	98.6	93.1	97.6	

B.2 DPTM Details

Method-related Hyperparameters and Configurations. We employ stable-diffusion v1-5 [23] as the diffusion model to generate 512×512 images with 20 denoising steps. $\gamma_1 = 5.5$ and $\gamma_2 = 0$. We set the threshold E to 0.01, and the total refinement iteration count R to 10. Note that setting E and R to other values may obtain superior performance.

Training-related Hyperparameters and Configurations. For the target model architecture, we adopt ResNet-101 for VisDA [21] and also for the other datasets. To further enhance the performance of the target model while mitigating the inevitable domain discrepancy between pseudo-target and real target domains, we incorporate the baseline Unsupervised Domain Adaptation (UDA) method BNM [7] during the target model training with pseudo-target data. This ensures that even when the samples generated by our method are not fully aligned with the real target distribution, the inconsistency can be further alleviated through the BNM adaptation process. For training hyperparameters, all parameters, including weight_decay, lr_gamma, lr_decay, momentum, n_iter_per_epoch, and n_epoch as 100, and lr remains unchanged as Source Model Pre-training in Section B.1.

Experiments Compute Resources. For Office-31 [25] and Office-Home [31], all related experiments are conducted using a single NVIDIA Tesla V100. For the large-scale Domain VisDA [21] and DomainNet-126 [20], all related experiments are conducted using a single NVIDIA Tesla H100.

C Supplementary Results

C.1 Full Results on VisDA

We present full results on VisDA in Table A.1. We report accuracy results over 12 categories and report the per-class accuracy. Notably, we also reproduced the results using only the source model. Experimental results demonstrate that our method outperforms existing SOTA methods across all categories. Moreover, our method demonstrates two significant advantages: (1) For categories such as plane, horse, knife, plant, and skateboard, our method achieves near-perfect classification accuracy, exceeding 99%. (2) For challenging categories like truck and car, where existing methods perform poorly, our approach substantially improves accuracy to over 90%. For the truck category, where most methods exhibit poor performance, our method exceeds the current SOTA method by 24.1%.

Manipulation Mechanism of Non-trust Set. We present ablation studies on different components of the Manipulation Mechanism of Non-trust Set via visualization results. As is demonstrated in Table 4, SDXL and SD15 present comparable results. To present better visualization, we employ SDXL

Table A.2: Full Results (%) on **VisDA** evaluated with ResNet-50.

Method	plane	bycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Perclass
Source-R50	79.7	35.5	44.5	63.8	62.0	25.0	86.9	26.6	77.6	30.0	94.7	12.7	53.3
Ours-R50	99.5	96.1	93.5	82.5	98.3	99.2	96.3	96.8	98.8	98.5	98.1	81	94.9

Table A.3: Comparison results with DATUM on Office-Home dataset.

Method	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg.
Source	50.1	67.9	74.4	55.2	65.2	67.2	53.4	44.5	74.1	64.2	51.5	78.7	62.2
DATUM	55.3	76.8	79.3	65.1	77.7	78.6	62.4	52.1	79.7	66.6	55.9	80.5	69.2
DPTM (ours)	86.7	94.2	92.8	91.5	94.0	92.6	90.6	86.4	92.8	90.5	87.1	94.7	91.2

Table A.4: Component-wise ablation studies on Office-Home dataset.

TGI	SFI	DFP	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg.
×	×	×	50.1	67.9	74.4	55.2	65.2	67.2	53.4	44.5	74.1	64.2	51.5	78.7	62.2
✓	×	×	69.2	86.2	82.2	74.6	87.8	80.7	76.7	67.8	82.4	73.5	66.9	87.1	77.9
×	✓	×	59.6	80.8	82.3	67.9	83.1	80.2	64.6	68.1	81.7	70.6	67.9	86.5	74.4
×	×	✓	67.8	87.6	82.7	68.7	84.0	80.6	70.7	67.6	82.4	74.1	67.1	88.7	76.8
×	✓	✓	75.9	89.4	88.3	78.9	87.9	87.0	76.0	74.1	87.8	80.0	75.9	89.1	82.5
✓	×	✓	72.9	88.0	84.1	77.5	88.3	83.6	76.8	69.0	84.0	75.7	68.8	88.4	79.8
✓	✓	×	70.2	89.9	85.5	81.1	89.4	89.3	80.7	70.7	86.3	80.3	72.1	91.0	82.2
✓	✓	✓	86.7	94.2	92.8	91.5	94.0	92.6	90.6	86.4	92.8	90.5	87.1	94.7	91.2

and set the denoising steps to 50. We present images generated by: (a) \mathbf{x}_l^u (b) $\tilde{\mathbf{x}}_l^u$ w/o Target-guided Initialization (c) $\tilde{\mathbf{x}}_l^u$ w/o Semantic Feature Injection (d) $\tilde{\mathbf{x}}_l^u$ w/o Domain-specific Feature Preservation (e) $\tilde{\mathbf{x}}_l^u$ of our method, respectively. As shown in Figure 2: (1) Our method’s manipulated samples $\tilde{\mathbf{x}}_l^u$, exhibit the best semantic alignment with their assigned labels $\hat{y}_l =$ and the best preservation of target distribution characteristics. (2) Column (b) (d) (e) that involve Semantic Feature Injection transforms the original semantics to the assigned label successfully, while Column (c) w/o Semantic Feature Injection achieves poor alignment with the assigned label, demonstrating the effectiveness of Semantic Feature Injection. (3) Column (b) excludes Target-guided Initialization, and only Domain-specific Feature Preservation works for maintaining the images within the target distribution. As a result, Column (b) preserves target domain features worse than Column (e). Similarly, Column (d) excludes Domain-specific Feature Preservation, and only Target-guided Initialization works, also exhibiting worse preservation of target domain features than Column (e). These results demonstrate the effectiveness of Target-guided Initialization and Domain-specific Feature Preservation.

The SFDA Model Size. Table A.1 shows results on VisDA with ResNet-101. To further demonstrate the superior performance of our method, we provide extra results with ResNet-50 in Table A.2. These results demonstrate that: (1) Our method exhibits robustness to model size. It maintains high performance even when using a smaller ResNet-50 target SFDA model. Notably, our method with ResNet-50 even outperforms existing comparative methods that use a larger ResNet-101 backbone, highlighting its superior adaptation performance regardless of model scale. Our method is also scalable with respect to the target SFDA model size. When using a larger ResNet-101 target SFDA model, our method achieves better performance compared to using ResNet-50, suggesting that its effectiveness can scale with increased target SFDA model size.

Different Components of DPTM. We conduct ablation studies on the independent usage of each individual component. For clarity in the tables, we refer to Target-guided Initialization, Semantic Feature Injection, and Domain-specific Feature Preservation as TGI, SFI, and DFP, respectively. We present comprehensive component-wise ablation results, including the performance of the model with only one component enabled and with each component individually removed. The results are shown in Table A.4, further demonstrating the effectiveness of our method.

Table A.5: The trust set accuracy evolving with r on the Office-Home dataset.

r	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr
1	95.4	97.6	98.5	99.1	100.0	99.8	100.0	100.0	99.8	100.0	97.4	99.1
2	85.3	95.0	97.2	88.8	93.9	96.1	85.9	83.8	96.6	87.5	86.4	96.3
3	94.5	98.3	97.6	91.2	98.2	97.3	90.0	91.9	97.8	92.9	93.1	98.2
4	94.2	98.0	98.1	94.3	98.1	97.8	93.6	95.2	98.0	94.4	95.7	98.4
5	96.6	99.0	98.7	95.1	99.0	98.5	94.4	95.8	98.2	95.6	95.7	99.0
6	96.3	99.1	98.6	95.7	99.1	98.6	96.1	96.8	98.6	95.0	97.3	99.0
7	97.1	99.4	98.5	96.9	99.4	98.5	97.1	96.8	98.7	96.8	96.6	99.2
8	97.2	99.3	99.2	97.0	99.2	99.2	96.6	97.9	98.8	95.7	97.2	99.5
9	98.6	99.5	99.2	97.0	99.6	99.2	97.0	98.1	98.9	97.3	97.8	99.3
10	97.5	99.4	98.7	97.5	99.6	99.1	97.2	97.7	99.4	97.3	98.5	99.5

Table A.6: The performance trajectory of setting all target samples to the non-trust set on the Office-Home dataset.

r	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar
0	50.1	67.9	74.4	55.2
1	56.4	78.7	81.7	63.0
2	55.9	78.5	81.4	62.8
3	55.9	78.3	81.6	62.7
4	55.8	78.1	81.4	62.7
5	55.7	77.9	81.3	62.7

D Supplementary Analyses

D.1 Analysis on Trust and Non-trust Partition for Target Domain.

Reliability of entropy-based trust and non-trust partition. Entropy-based selection of reliable pseudo-labels of target samples is commonly used in SFDA, and its effectiveness is demonstrated in prior studies [16]. To further evaluate its reliability, we report the trust set accuracy evolving with r on the Office-Home dataset as Table A.5 (r denotes the r -th refinement iteration, where $r = 1, 2, \dots, R$, and in our experiments we set $R = 10$). Figure 2(a) shows that the size of the trust set grows as r increases. Remarkably, the trust accuracy remains consistently high across all tasks using a total of 10 refinement iterations. Besides, when r is small, the trust set accuracy may not be high, but will increase to a high value as r grows. For example, in tasks like Pr→Ar and Pr→Cl, trust accuracy is below 90% when $r = 2$ but reaches higher than 97% when $r = 10$. These results validate the effectiveness of our method, which could correct errors in the trust set with the growth of r . Note that the non-trust set accuracy does not affect the performance of our method, as we completely discard the original pseudo-labels of non-trust samples.

Set all target samples to the non-trust set. The high performance gain of our method stems from two key mechanisms: i) progressively expanding the trust set with high-accuracy pseudo-labels to allow the SFDA model to learn real target domain features, and ii) progressively reducing the manipulated non-trust set. Mechanism ii) is critical to prevent features from the synthetic domain from becoming dominant, since there is an inherent gap that persists between the synthetic and real target domains (even though we employ alignments to bridge the gap). This phenomenon fundamentally stems from the inherent domain gap between synthetic and real data, a well-documented challenge that has been rigorously demonstrated in prior work [1]. Therefore, canceling the trust set could cause degraded performance, since we could only obtain features from the synthetic domain. We validate it with an empirical study on the first 4 Office-Home tasks (Ar→Cl, Ar→Pr, Ar→Rw, and Cl→Ar), where we cancel the trust set and set all target samples to the non-trust set. r denotes the r -th refinement iteration, and we report results of $r = 1, 2, \dots, 5$. As shown in Table A.6, when using only non-trust samples, the performance is not improved as r grows.

Table A.7: Comparison between Random assignment of labels and using original pseudo-labels for non-trust samples.

Method	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar
Original pseudo-labels	80.3	91.1	90.1	83.9
Ours (randomly assigned labels)	86.7	94.2	92.8	91.5

D.2 Analysis on Manipulation of Non-trust Set.

Random assignment of labels for non-trust samples. From a performance perspective, we employ random label assignment, instead of using the original pseudo-labels of non-trust samples given by the SFDA model, for two key reasons. First, there are no obvious patterns between the original pseudo-labels and ground truth labels of non-trust samples, and the original pseudo-labels cannot provide valid semantic priors. Second, if we use the diffusion model to semantically transform non-trust samples to their original pseudo-labels, this approach could inevitably introduce the class imbalance issue in the manipulated non-trust set. Therefore, we adopt a uniform random label reassignment strategy as formalized in Equation (2). To validate this claim, we perform comparative experiments using the original pseudo-labels for generation. Due to time limits, the experiments were performed on the first 4 Office-Home tasks (Ar→Cl, Ar→Pr, Ar→Rw, and Cl→Ar). The results presented in Table A.7 demonstrate the validity of our choice. Besides, from the perspective of reproducibility and stability, the results are fully reproducible as long as the random seed is fixed. In this case, the division between trust and non-trust sets becomes repeatable, and the sample ordering in the non-trust set’s data loader is reproducible. Consequently, the new labels assigned to non-trust samples via Equation (2) are perfectly reproducible.

D.3 Analysis of Comparison with DATUM

We compare our method with DATUM [3] on the Office-Home dataset. According to the DATUM [3] paper, the method consists of three stages: (1) Employing training of DreamBooth [24] to personalize the diffusion model by associating a unique token V_* with the appearance of the target domain. (2) Using the personalized diffusion model to generate a pseudo-target domain. (3) Training an existing UDA framework on the labeled source data and the generated unlabeled pseudo-target data. To align DATUM with the SFDA setting, we modify only the third stage. Specifically, we first train a source model using labeled source data, and then adapt the model to the target domain using the pseudo-target data generated by DATUM.

The results are shown in Table A.3. For reference, we also include the performance of the source model as a baseline for comparison. The results demonstrate that:

(1) Compared with the source model, DATUM achieves better adaptation performance, demonstrating its effectiveness in the SFDA setting. This result suggests that diffusion-based domain adaptation methods like DATUM are capable of significantly improving adaptation performance. We consider DATUM a valuable and insightful work, as its design showcases the potential of leveraging diffusion models to generate pseudo-target data for SFDA.

(2) Compared with our method, the results demonstrate that DATUM performs significantly worse than our method. This may be due to the following two reasons:

- A key factor contributing to the superior performance of our method is the use of our Progressive Refinement Mechanism, which enables the SFDA model to progressively improve its performance through multiple iterations. In contrast, DATUM lacks such a dynamic update mechanism, which may limit its adaptation ability.
- We provide the detailed performance trajectory of our method as r increases from 1 to 10 in Table 6. Notably, when $r = 1$, our method still outperforms DATUM. This demonstrates that even without the Progressive Refinement Mechanism, our method remains more effective than DATUM.

(3) The aforementioned results further indicate that the pseudo-target domain generated by our method is better aligned with the real target domain compared to that generated by DATUM. We attribute this to the following reason. DATUM relies on DreamBooth to learn the appearance of the

Table A.8: Unsupervised Model Selection results on the VLCS dataset.

Method	UMS	C→L	C→S	C→V	L→C	L→S	L→V	S→C	S→L	S→V	V→C	V→L	V→S	Avg
No adapt	✗	47.8	53.3	64.6	51.3	40.7	55.1	58.1	36.4	55.4	97.8	48.8	72.0	56.7
SHOT [15]	✗	44.6	55.7	75.9	65.3	60.8	74.8	87.7	41.1	82.7	89.8	45.8	65.7	65.6
SHOT++ [17]	✗	41.7	58.4	75.6	70.0	56.9	76.2	71.6	39.5	80.7	96.9	43.7	60.07	64.3
AaD [32]	✗	37.7	57.2	75.5	59.5	48.8	67.5	84.0	34.8	72.8	43.4	40.2	54.8	56.4
CoWA-JMDS [13]	✗	46.1	58.4	81.1	85.6	64.1	78.2	95.9	48.1	82.7	99.4	50.8	65.3	71.3
NRC [33]	✗	39.9	55.9	75.5	64.7	54.1	74.8	78.4	40.3	82.2	90.1	41.7	62.8	63.4
G-SFDA [34]	✗	42.6	54.9	73.5	82.3	51.4	72.0	74.6	45.8	82.7	88.9	49.1	64.3	65.2
FT [14]	✗	50.1	66.7	81.1	99.7	62.2	78.0	99.8	55.5	80.5	99.7	53.0	67.0	74.5
LP-FT [14]	✗	50.5	66.7	79.5	99.7	65.6	77.6	99.7	54.1	79.5	99.7	51.2	69.8	74.5
DPTM (Ours)	✓	88.2	96.0	90.8	97.7	97.2	99.2	99.6	88.6	89.4	99.9	92.7	91.5	94.2

target domain and map it to a unique token V_* . However, for classification tasks, this process becomes challenging when the true class labels of the target data are unknown. According to the paper of DATUM, during DreamBooth training, the lack of class labels forces the use of vague prompts such as "a photo of a V_* object". This may cause the learned token to not only capture the domain-specific appearance of the target data, but also absorb semantic information related to the object class and even irrelevant background features. As a result, the generated pseudo-target images may exhibit limited alignment with the true target distribution.

D.4 Unsupervised Model Selection

For more complex scenarios, we can also explore other hyperparameter values. In such cases, unsupervised model selection becomes necessary. We employ an unsupervised model selection strategy using the nuclear norm. We extract the softmax output matrix for all target samples and calculate its nuclear norms. Then, we select the model with the largest nuclear norm [8, 9]. We train models with $E \in [0.01, 0.005, 0.001]$ and R from 1 to 10 on real-world benchmark VLCS [11] and apply this model selection metric. The results are shown in Table A.8, comparison results are from [14], where UMS is unsupervised model selection for short. Note that all the methods for comparison report their best target accuracy (according to the labels), while ours performs model selection without knowing target accuracy.

Besides, we also conduct unsupervised model selection experiments on the TerraIncognita dataset [2]. The experimental setting follows that of the VLCS dataset, and we also use the nuclear norm to select the best model without access to target domain labels. The comparison results are also taken from [14].

The results are shown in Table A.9. The results demonstrate that, similar to VLCS, our method significantly outperforms all compared methods even when using nuclear norm as the criterion for unsupervised model selection, achieving an average accuracy improvement of 12.4% over the best-performing comparative method. Notably, in some challenging scenarios such as L100→L43 and L38→L43, where existing methods generally perform poorly, our method achieves substantial gains of 34.8% and 36.5%, respectively, even under unsupervised model selection. It is worth noting that all compared methods were evaluated using their best-performing models selected based on target domain accuracy. These results further demonstrate the effectiveness and practicality of our method.

We also acknowledge that there may exist better criteria beyond the nuclear norm. Exploring more effective model selection metrics will be an important direction for our future work, as we believe that a more suitable criterion could further unleash the potential of our method and enhance its practical applicability.

D.5 Visualization

Feature Visualization. We visualize the feature distribution of our DPTM on the $Rw \rightarrow Cl$ task of the Office-Home dataset using t-SNE, where other SFDA methods perform poorly. The visualization

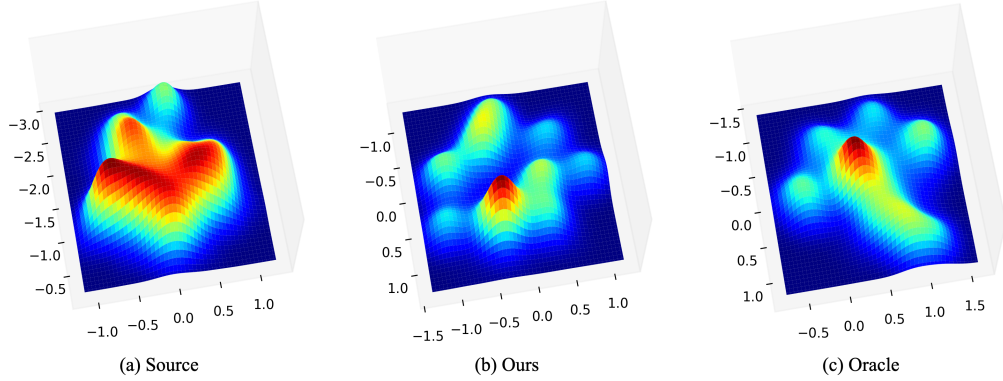


Figure A.1: Feature distribution visualization on the $Rw \rightarrow Cl$ task of the Office-Home dataset.

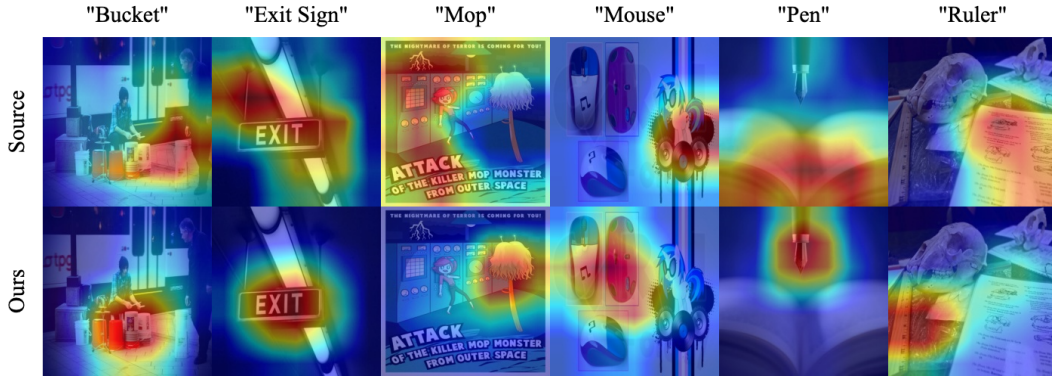


Figure A.2: Grad-CAM visualization on the Office-Home dataset.

results are shown in Figure A.1. We compare the results with **Source** and **Oracle**, where **Source** denotes the source model trained only on Rw , and **Oracle** represents the real target model trained on Cl with ground-truth labels. The visualization is presented as 3D density charts. Following DIFO [29], we use the first 10 categories for clearer viewing.

Grad-CAM Visualization. We further perform Grad-CAM visualization on the Office-Home dataset to analyze the attention behavior of our model. As shown in Figure A.2, compared with the source model, our method consistently focuses on the target objects, while the source model tends to attend to irrelevant background regions or even the entire image.

E Time and Memory Costs

E.1 Theoretical Analysis

E.1.1 Time Cost

The time cost can be mainly divided into two parts:

- **SFDA model training.** For any r -th refinement iteration with $r = 1, 2, \dots, R$, we train the SFDA model using the trust set and the manipulated non-trust set. The total number of samples of these two sets equals that of the target domain. We use supervised training here. The time cost mainly depends on the scale of the target domain, and remains constant as r grows. We denote it as t_{SFDA} .
- **Sample generation.** For any r -th iteration refinement with $r = 1, 2, \dots, R$, the generation time for each sample remains constant, at about 1 second on a single NVIDIA Tesla V100. The generation time depends on the number of samples of the non-trust set, which is a subset of the target domain. Note that, with the growth of r , the size of the non-trust set declines

Table A.9: Unsupervised Model Selection results on the TerraIncognita dataset.

Method	UMS	L100→L38	L100→L43	L100→L46	L38→L100	L38→L43	L38→L46	L43→L100	L43→L38	L43→L46	L46→L100	L46→L38	L46→L43	Avg
No adapt	✗	26.2	20.3	27.1	29.3	31.4	31.6	24.1	44.1	38.7	33.6	21.6	22.2	29.2
SHOT [15]	✗	20.1	23.8	28.5	36.0	29.0	13.6	26.2	14.5	32.7	34.3	12.6	37.4	25.7
SHOT++ [17]	✗	29.3	22.1	25.5	22.8	31.8	18.4	33.3	22.6	25.6	35.8	13.0	44.6	27.1
AaD [32]	✗	17.2	17.4	22.1	24.6	28.1	13.3	28.9	23.3	23.1	31.6	7.4	34.6	22.6
CoWA-JMDS [13]	✗	33.1	31.4	26.4	36.3	38.3	19.3	28.2	13.6	26.6	32.5	10.0	47.6	28.7
NRC [33]	✗	19.3	22.7	29.5	38.5	26.9	14.9	30.8	22.6	32.2	28.9	11.0	39.0	26.4
G-SFDA [34]	✗	21.6	29.1	38.2	38.4	27.0	22.4	40.9	17.4	33.3	35.0	16.3	52.6	31.0
FT [14]	✗	52.4	41.7	50.0	63.8	38.6	47.8	66.2	56.7	51.4	68.9	56.7	61.2	54.6
LP-FT [14]	✗	54.3	47.5	46.9	63.6	41.3	49.0	64.2	55.9	54.4	68.4	55.7	63.8	55.4
DPTM (ours)	✓	61.3	82.3	58.3	74.7	77.8	50.2	69.1	66.1	58.3	75.1	63.6	77.0	67.8

rapidly, as shown in Figure 2. Thus, the total generation time for R iterations equals the total number of generated samples N_{total} .

The total training time t can be estimated by $t = R * t_{SFDA} + N_{total}$. We take the 12 tasks in the Office-Home dataset shown in Table 1 as examples. We set $E = 0.01$ and $R = 10$ and run the tasks on a single NVIDIA Tesla V100. It takes about 7.4 hours for SFDA model training and 3.8 hours for generation to complete the total algorithm on average for each task, demonstrating an acceptable time cost. Note that our model also allows parallel training on multiple GPUs to further reduce the time cost.

E.1.2 Memory Cost

The memory needs can be mainly divided into two parts:

- **SFDA Model Training.** During any r -th refinement iteration, the memory needs remain constant and approximately equal to those of standard supervised learning on the target domain, and will not accumulate as r grows.
- **Sample Generation.** Our method generates samples sequentially (one sample at a time). Thus, this part only requires sufficient memory to run the Stable Diffusion model.

E.2 Experimental Evaluations

We compiled comparative experiments on cost analysis, including training time and peak GPU memory usage for our method and other benchmark methods whose official code is publicly available and runnable. We first describe the settings of our cost analysis experiments.

Datasets. We conduct experiments on the Office-Home dataset (moderate scale) and the DomainNet-126 dataset (large scale), as both contain a sufficient number of samples to reasonably evaluate the computational efficiency of different methods.

Benchmarked methods. We successfully ran the following benchmark methods: SHOT, NRC, GKD, AdaCon, CoWA, SCLM, PLUE, TPDS, DIFO, and ProDe. All methods were executed strictly following the instructions provided in their official code repositories.

Measurement protocol. Due to time constraints, we ran each comparative method for one epoch per SFDA task, recorded the training time and peak GPU memory usage during that epoch, and multiplied the time by the number of total epochs provided in their official code repositories to estimate the full training time. As for our own method, we had detailed logs from prior experiments, and we report the actual training time and GPU memory usage based on our full training runs. And we ran all the tasks with a single NVIDIA Tesla V100.

Training time (hours) on the Office-Home dataset is listed in Table A.10. Office-Home is a medium-scale dataset that contains 4 domains, including Art (Ar, 2427 images), Clipart (Cl, 4365 images), Product (Pr, 4439 images), and Real-World (Rw, 4357 images).

Similarly, training time (hours) on the DomainNet-126 dataset is listed in Table A.11. DomainNet-126 is a large-scale dataset that contains 4 domains, including clipart (C, 18523 images), painting (P, 10212 images), real (R, 69622 images), and sketch (S, 24147 images).

As for inference time, following most of the existing benchmark methods, our method uses only the target model (e.g., ResNet-50) for inference on the target domain. No additional modules or auxiliary

Table A.10: Training time (hours) on the Office-Home dataset.

Method	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg.
SHOT	3.0	2.8	8.2	1.8	3.0	8.3	1.5	2.7	7.8	1.4	2.9	3.0	3.9
NRC	0.9	1.4	2.4	0.9	1.2	3.9	1.3	1.7	2.6	1.0	1.2	1.4	1.7
GKD	2.7	2.9	8.3	1.6	2.8	8.4	1.6	3.0	7.6	1.6	2.9	2.6	3.8
AdaCon	0.2	0.2	0.5	0.1	0.2	0.4	0.2	0.2	0.4	0.1	0.2	0.2	0.2
CoWA	0.9	2.4	3.5	0.6	1.2	2.6	0.8	1.6	1.5	1.0	1.2	1.5	1.6
SCLM	2.9	3.2	7.9	1.5	3.0	9.2	1.6	2.8	8.6	1.5	2.8	2.8	4.0
PLUE	0.2	0.3	0.6	0.2	0.5	0.5	0.1	0.2	0.4	0.2	0.2	0.2	0.3
TPDS	12.3	2.7	8.0	1.6	2.9	8.4	1.5	3.0	7.8	1.4	6.0	7.5	5.3
DIFO	6.9	7.2	22.4	4.1	7.3	25.5	4.3	7.7	26.5	4.4	7.3	7.8	11.0
ProDe	2.6	2.5	3.6	1.6	2.8	3.2	1.4	2.9	3.8	1.5	2.1	2.8	2.6
DPTM (ours)	11.1	9.7	16	8.7	9.9	16.2	8.6	11.3	16.2	8.6	11.1	9.7	11.4

Table A.11: Training time (hours) on the DomainNet-126 dataset.

Method	C→P	C→R	C→S	P→C	P→R	P→S	R→C	R→P	R→S	S→C	S→P	S→R	Avg.
SHOT	10.9	392.8	51.5	34.9	437.9	68.4	37.3	14.2	66.4	37.5	12.3	391.7	129.7
NRC	1.0	5.4	2.3	3.3	9.5	4.6	2.1	1.5	3.9	2.9	1.1	4.8	3.5
GKD	12.2	4530.7	59.7	35.3	4134.7	67.1	37.3	10.4	72.7	38.6	13.1	4019.8	1086.0
AdaCon	0.5	2.3	0.9	0.7	2.8	0.9	0.7	0.4	0.9	0.7	0.4	2.8	1.2
CoWA	1.4	6.2	5.3	4.8	7.8	5.4	4.4	2.8	6	3.2	2.1	9.3	4.9
PLUE	0.5	2.8	1.0	0.8	2.8	4.6	0.8	0.5	1.0	0.8	0.5	2.8	1.6
TPDS	11.3	383.4	52.7	39.3	426.9	68.3	39.4	14.1	69.2	40.2	13.9	406.3	130.4
DIFO	26.0	918.4	116.0	79.9	960.7	146.5	79.0	29.1	116.9	67.2	29.0	935.4	292.0
ProDe	5.1	27.1	17.0	15.8	48.1	16.7	8.8	8.1	16.9	7.8	5.2	28.5	17.1
DPTM (Proposed)	13.7	45.3	28.5	22.2	43.6	30.1	22.2	13.7	24.8	20.4	13.3	43.5	26.8

Table A.12: The peak GPU memory usage.

Method	Office-Home	DomainNet-126
SHOT	7GBytes	7GBytes
NRC	5GBytes	7GBytes
GKD	7GBytes	8GBytes
AdaCon	14GBytes	14GBytes
CoWA	7GBytes	8GBytes
SCLM	7GBytes	-
PLUE	13GBytes	14GBytes
TPDS	7GBytes	7GBytes
DIFO	7GBytes	11GBytes
ProDe	12GBytes	13GBytes
DPTM (Proposed)	10GBytes	12GBytes

models are involved during inference, so the inference time of our method is effectively the same as that of other benchmark methods using the same backbone.

Besides, the peak GPU memory usage is listed in Table A.12.

F Limitation

Since we employ the pre-trained Stable Diffusion model [23], the performance of DPTM is inherently constrained by Stable Diffusion’s generation capabilities. For the SFDA benchmarks used in our experiments, which primarily consist of common object categories that Stable Diffusion can reliably generate, the framework achieves strong performance. However, when dealing with custom datasets containing categories that are challenging for Stable Diffusion to generate (e.g., specialized medical instruments or rare industrial components), it may be necessary to first collect relevant training data for these specific classes and fine-tune Stable Diffusion accordingly.

References

- [1] Amila Akagic, Emir Buza, Medina Kapo, and Mahdi Bohlouli. Exploring the impact of real and synthetic data in image classification: A comprehensive investigation using cifake dataset. In *2024 10th International Conference on Control, Decision and Information Technologies (CoDIT)*, pages 1207–1212. IEEE, 2024.
- [2] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pages 456–473, 2018.
- [3] Yasser Benigmim, Subhankar Roy, Slim Essid, Vicky Kalogeiton, and Stéphane Lathuilière. One-shot unsupervised domain adaptation with personalized diffusion models. In *Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 698–708, 2023.
- [4] Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. In *Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 295–305, 2022.
- [5] Shivang Chopra, Suraj Kothawade, Houda Aynaou, and Aman Chadha. Source-free domain adaptation with diffusion-guided source data generation. *arXiv preprint arXiv:2402.04929*, 2024.
- [6] Chaoran Cui, Fan’an Meng, Chunyun Zhang, Ziyi Liu, Lei Zhu, Shuai Gong, and Xue Lin. Adversarial source generation for source-free domain adaptation. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(6):4887–4898, 2023.
- [7] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Liang Li, Qingming Huang, and Qi Tian. Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3941–3950, 2020.
- [8] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Liang Li, Qingming Huang, and Qi Tian. Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3941–3950, 2020.
- [9] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Liang Li, Qingming Huang, and Qi Tian. Fast batch nuclear-norm maximization and minimization for robust domain adaptation. *arXiv preprint arXiv:2107.06154*, 2021.
- [10] Yuntao Du, Haiyang Yang, Mingcai Chen, Hongtao Luo, Juan Jiang, Yi Xin, and Chongjun Wang. Generation, augmentation, and alignment: A pseudo-source domain based method for source-free domain adaptation. *Machine Learning*, 113(6):3611–3631, 2024.
- [11] Chen Fang, Ye Xu, and Daniel N Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1657–1664, 2013.
- [12] Jonghyun Lee, Dahuin Jung, Junho Yim, and Sungroh Yoon. Confidence score for source-free unsupervised domain adaptation. In *Proceedings of the 39th International Conference on Machine Learning*, pages 12365–12377, 2022.
- [13] Jonghyun Lee, Dahuin Jung, Junho Yim, and Sungroh Yoon. Confidence score for source-free unsupervised domain adaptation. In *International conference on machine learning*, pages 12365–12377. PMLR, 2022.
- [14] Suho Lee, Seungwon Seo, Jihyo Kim, Yejin Lee, and Sangheum Hwang. Few-shot fine-tuning is all you need for source-free domain adaptation. *arXiv preprint arXiv:2304.00792*, 2023.
- [15] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *Proceedings of the 37th International Conference on Machine Learning*, pages 6028–6039, 2020.
- [16] Jian Liang, Dapeng Hu, Yunbo Wang, Ran He, and Jiashi Feng. Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8602–8617, 2021.
- [17] Jian Liang, Dapeng Hu, Yunbo Wang, Ran He, and Jiashi Feng. Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8602–8617, 2021.
- [18] Mattia Litrico, Alessio Del Bue, and Pietro Morerio. Guiding pseudo-labels with uncertainty estimation for source-free unsupervised domain adaptation. In *Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7640–7650, 2023.

- [19] Yu Mitsuzumi, Akisato Kimura, and Hisashi Kashima. Understanding and improving source-free domain adaptation from a theoretical perspective. In *Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 28515–28524, 2024.
- [20] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1406–1415, 2019.
- [21] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017.
- [22] Zhen Qiu, Yifan Zhang, Hongbin Lin, Shuaicheng Niu, Yanxia Liu, Qing Du, and Mingkui Tan. Source-free domain adaptation via avatar prototype generation and adaptation. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI-21)*, pages 2921–2927, 2021.
- [23] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022.
- [24] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dream-booth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22500–22510, 2023.
- [25] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *Proceedings of the 11th European Conference on Computer Vision*, pages 213–226, 2010.
- [26] Song Tang, An Chang, Fabian Zhang, Xiatian Zhu, Mao Ye, and Changshui Zhang. Source-free domain adaptation via target prediction distribution searching. *International Journal of Computer Vision*, 132(3):654–672, 2024.
- [27] Song Tang, Yuji Shi, Zhiyuan Ma, Jian Li, Jianzhi Lyu, Qingdu Li, and Jianwei Zhang. Model adaptation through hypothesis transfer with gradual knowledge distillation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5679–5685, 2021.
- [28] Song Tang, Wenxin Su, Yan Gan, Mao Ye, Jianwei Dr. Zhang, and Xiatian Zhu. Proxy denoising for source-free domain adaptation. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [29] Song Tang, Wenxin Su, Mao Ye, and Xiatian Zhu. Source-free domain adaptation with frozen multimodal foundation model. In *Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23711–23720, 2024.
- [30] Song Tang, Yan Zou, Zihao Song, Jianzhi Lyu, Lijuan Chen, Mao Ye, Shouming Zhong, and Jianwei Zhang. Semantic consistency learning on manifold for source data-free unsupervised domain adaptation. *Neural Networks*, 152:467–478, 2022.
- [31] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 5018–5027, 2017.
- [32] Shiqi Yang, Shangling Jui, Joost Van De Weijer, et al. Attracting and dispersing: A simple approach for source-free domain adaptation. In *Advances in Neural Information Processing Systems 35*, pages 5802–5815, 2022.
- [33] Shiqi Yang, Joost Van de Weijer, Luis Herranz, Shangling Jui, et al. Exploiting the intrinsic neighborhood structure for source-free domain adaptation. In *Advances in Neural Information Processing Systems 34*, pages 29393–29405, 2021.
- [34] Shiqi Yang, Yaxing Wang, Joost Van De Weijer, Luis Herranz, and Shangling Jui. Generalized source-free domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8978–8987, 2021.
- [35] Li Yi, Gezheng Xu, Pengcheng Xu, Jiaqi Li, Ruizhi Pu, Charles Ling, Ian McLeod, and Boyu Wang. When source-free domain adaptation meets learning with noisy labels. In *The Eleventh International Conference on Learning Representations*, 2023.
- [36] Lihua Zhou, Nianxin Li, Mao Ye, Xiatian Zhu, and Song Tang. Source-free domain adaptation with class prototype discovery. *Pattern Recognition*, 145:109974, 2024.